



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/902,469	07/10/2001	David S. Ebbo	160315.01	6779
22971	7590	10/14/2005	EXAMINER STEELMAN, MARY J	
MICROSOFT CORPORATION ATTN: PATENT GROUP DOCKETING DEPARTMENT ONE MICROSOFT WAY REDMOND, WA 98052-6399			ART UNIT 2191	PAPER NUMBER

DATE MAILED: 10/14/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/902,469	EBBO ET AL.
	Examiner Mary J. Steelman	Art Unit 2191

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 03 August 2004.

2a) This action is FINAL. 2b) This action is non-final.

3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 1-19 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5) Claim(s) _____ is/are allowed.

6) Claim(s) 1-19 is/are rejected.

7) Claim(s) _____ is/are objected to.

8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.

10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
a) All b) Some * c) None of:
1. Certified copies of the priority documents have been received.
2. Certified copies of the priority documents have been received in Application No. _____.
3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) Notice of References Cited (PTO-892)
2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date 3 August 2004

4) Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.
5) Notice of Informal Patent Application (PTO-152)
6) Other: _____

DETAILED ACTION

1. This Office Action is in response to RCE and IDS received 3 August 2004. The prior Allowance has been withdrawn. Prosecution has been reopened. Claims 1-6 and 8-20 are pending. There is no claim 7. The pending claims have been renumbered as 1-19 per 37 CFR 1.126.

Information Disclosure Statement

2. IDS received 3 August 2004 has been considered.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 1-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over US Patent 5,835,712 to DuFresne, in view of US PreGrant Pub 20030074634 A1 to Emmelmann.

Per claim 1:

In a server computer system having memory, a method...by the server computer system to create server-side objects for dynamically rendering web page content, the web page content delivered to a client-side computer system and displayed as a web page on the client computer system, said method comprising:

(DuFresne: Col. 2, lines 58-63, "a design, development, maintenance, and deployment systems and methods in a client server hypertext environment...dynamic client server environment..." (emphasis added), col. 4, lines 37-39, "The resulting source (create server side objects) to the displayable page (content delivered to a client-side computer system and displayed as a web page) is...a pure HTML document which can be interpreted by the client browser...")

-receiving a request from the client specifying a dynamic web page content file;

(DuFresne: Col.3, lines 11-12, "in response to a request for the Web page from a client...")

-processing the dynamic web page content file to produce a source code file containing source code that represents control objects declared in the web page content file;

(DuFresne: Col. 47-54, "...in a hypertext source of the present invention, a value associated with a tag extension can be a script such that processing the source (processing the dynamic web page content file) executes the tag extension and further causes the script to expand and execute a string of tags and instructions (control objects). Upon exhaustion of the extension and/or instructions, the resulting static value replaces the original tag extension in the source (produce source code file containing source code that represents control objects declared).")

-compiling the source code file...from which a set of hierarchical objects can be instantiated to produce web page authoring language that produces a web page for display;

(DuFresne: Col. 3, lines 41-52, "...instructions can be nested and grouped (hierarchical objects) to form a script which performs predefined tasks..." Source is processed by processor to expand

the tags (compile source code / transform to language that can be used to author a web page) and produce a source code that may be displayed as a web page by the client (col. 4, lines 32-38).

-wherein the source code file declaratorily refers to one or more additional dynamic web page content files, each reference to the one or more additional dynamic web page content files correspond to a single hierarchical object within the set of hierarchical objects.

(DuFresne: Col. 7, lines 5-16, "An important aspect of the HTML is creating hypertext links to connect one Web page to another (declaratorily refer to one or more additional dynamic web page content files)...links are also created using tags...code for directing (correspond to a single hierarchical object within a set of hierarchical objects / use directory path or URL) the browser to a certain file...")

DuFresne failed to disclose specific details regarding 'creating a class'. However Emmelmann suggested component classes [0142] & [0189]. Therefore, it would have been obvious, to one of ordinary skill in the art to consider the components (col. 8, line 60) of the forms as disclosed by DuFresne to be of a class format, as classes are well known and commonly used in HTML programming. Both inventions provide dynamic client/server environments (DuFresne, col. 2, line 63 & Emmelmann, [0001]) and attempt to simplify (DuFresne, col. 2, lines 65-66 & Emmelmann, [0022]) the updating/modifying of dynamic aspects of components used in HTML.

Per claim 2:

-the source code file may be referenced by a second source code file as one of the one or more additional dynamic web page content files.

(DuFresne: Col. 7, lines 5-6, "creating hypertext links to connect one Web page to another..." Source code at a URL may be referenced by a hypertext link by a second source code when content is dynamic.)

Per claim 3:

-the dynamic web page content file is a server-side declaration datastore.

(DuFresne: Col. 4, lines 4-8, « Templates are stored in a template database. The field names and corresponding contents of a template are stored in a content database (datastore).")

Per claims 4 and 5:

-is stored in...memory on the server computer system and is available to instantiate objects in response to another request specifying the dynamic web page content file.

(DuFresne: Col. 3, lines 41-59, "instructions can be indexed and grouped to form a script which performs predefined tasks...a script is embedded in a tag extension...executing the extension causes the script to expand...a value associated with a tag extension can be a script...The present invention can provide one or more databases...for storing (store in cache memory and available to instantiate objects in response to request) the data values...")

DuFresne failed to disclose specific details regarding 'creating a class' and specifically storing in cache memory / on a magnetic medium. However Emmelmann suggested component classes

[0142] & [0189]. DuFresne does disclose storing, however fails to explicitly suggest storage in cache / on a magnetic medium. Storing in cache memory or on a magnetic medium is well known in the art. Therefore, it would have been obvious, to one of ordinary skill in the art to consider the components (col. 8, line 60) of the forms as disclosed by DuFresne to be of a class format, as classes are well known and commonly used in HTML programming. Both inventions provide dynamic client/server environments (DuFresne, col. 2, line 63 & Emmelmann, [0001]) and attempt to simplify (DuFresne, col. 2, lines 65-66 & Emmelmann, [0022]) the updating/modifying of dynamic aspects of components used in HTML. It would have been obvious to one of ordinary skill in the art, at the time of the invention to store (DuFresne: col. 4, lines 4-8, store in memory) in cache memory or on a magnetic storage medium, as it is commonly used for faster access to a memory location or for portability of an executing program.

Per claim 6:

-the step processing the dynamic web page content file comprises:
-parsing the dynamic web page content file to store portions of the file into a data model, the data model comprises a plurality of data objects linked in a hierarchical manner; (DuFresne: Col. 4, lines 32-38, "The source code defined within the template is then processed (processing the dynamic web page content file) by the processor. Such process includes executing the tag extensions in the source to replace each extension (parse content file, find tag extensio9ns) with the corresponding value so as to configure the page with the remaining

hypertext codes (store into HTML data model). The resulting source to the displayable page is, therefore, a pure HTML document...")

-generating source code related to declaration information based on an analysis of the data model during a first phase;

-writing the source code related to declaration information to the source code file;

(DuFresne: Col. 3, lines 36-40, "...a tag extension in a source (analysis of the data model) to a Web page is associated with a value in a database. Such a value can be static data or a variable, such as another tag extension (related to declaration information)", col. 3, lines 50-52, "processing (generating / writing) the source executes the tag extension and further causes the script to expand and execute a string of tags and instructions...")

-generating source code related to control object information based on an analysis of the data model during a second phase;

-writing the source code related to control object information to the source code file during the second phase;

(DuFresne: Col. 3, lines 36-40, "...a tag extension in a source (analysis of the data model) to a Web page is associated with a value in a database. Such a value can be static data or a variable, such as another tag extension. A value can also be an instruction or a group of instructions (related to control object information)", col. 3, lines 50-52, "processing (generating / writing) the source executes the tag extension and further causes the script to expand and execute a string of tags and instructions...")

-where the one or more of the plurality of data objects correspond to compiled versions of one or more additional dynamic web page content file.

(DuFresne: Expanding the code (plurality of data objects), executing the tag extensions transforms (compiles) the code (to corresponding version of one or more additional dynamic web page content file). Col. 3, lines 16-19, "extensions (including scripts, tag extensions, and instructions) are processed and replaced by the corresponding values (in memory)...")

DuFresne fails to suggest that code related to declaration information is processed during a first phase, while code related to control object information is processed during a second phase.

However, he does suggest that these types of code are processed, expanded, and (col. 3, lines 52-54) "the resulting static value replaces the original tag extension in the source (generating / writing source code to the source code file)." It would be obvious that both actions are done, and it is insignificant as to whether one action precedes the other.

Per claim 7:

- the method further comprises:
- generating source code related to rendering information based on an analysis of the data model during a third phase;
- writing the source code related to rendering information to the source code file during the third phase.

Art Unit: 2191

(DuFresne: Col. 9, lines 33-34, "This process produces an output HTML page (generating / writing source code related to rendering information based on an analysis of the data model) which is displayable on the client browser.")

Per claim 8:

- prior to the step of processing the dynamic web page content file, determining whether the class related to the received request has been compiled and stored in memory;
- if the class has been compiled and stored in memory, skipping the processing step, otherwise continue with the processing step.

(DuFresne: Col. 9, lines 30-32, "The template is processed by the processor and each code is replaced by a stored data value (related class is compiled and stored in memory) or executed to perform certain task (if not stored continue with processing to compile / expand code)...")

DuFresne failed to specifically disclose a sequence that first checked memory to determine whether the related class had been compiled and stored. However, he did disclose the two steps. Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify DuFresne's invention, to include explicit details regarding the steps, because DuFresne did disclose retrieving from memory else compile / expand code, and it only is logical to first determine whether stored code exists before continuing to process.

Per claim 9:

A computer data signal embodied in a carrier wave by a computing system having

memory and encoding a computer program for executing a computer process... creating a...memory, ...create server-side objects for dynamically rendering web page content, the web page content delivered to a client-side computer system and displayed as a web page on the client computer system, said computer process comprising:

(DuFresne: Col. 2, lines 58-59, "a design, development, maintenance, and deployment systems and methods...running in an HTTP) network environment (data signal embodied in a carrier wave by a system)...", col. 3, lines 4-6, "deploying client-server applications involves inserting executable tags in an HTML source (dynamically rendering web page content) to a displayable page (delivered & displayed on client)", col. 6, lines 22-26, "...server manages information stored (in memory) ...runs a program to manage HTML files (in memory)...")

- receiving a request from the client specifying a dynamic web page content file;
- processing the dynamic web page content file to produce a source code file containing source code that represent control objects declared in the web page content file;
- compiling the source code file to produce a class from which a set of hierarchical objects can be instantiated to produce web page authoring language that produces a web page for display;
- wherein dynamic web page content file declaratorily refers to one or more additional dynamic web page content files, each reference to the one or more additional web page content files correspond to a single hierarchical object within the set of hierarchical objects.

See rejection of limitations in claim 1 above.

Art Unit: 2191

DuFresne failed to disclose specific details regarding 'creating a class'. However Emmelmann suggested component classes [0142] & [0189]. Therefore, it would have been obvious, to one of ordinary skill in the art to consider the components (col. 8, line 60) of the forms as disclosed by DuFresne to be of a class format, as classes are well known and commonly used in HTML programming. Both inventions provide dynamic client/server environments (DuFresne, col. 2, line 63 & Emmelmann, [0001]) and attempt to simplify (DuFresne, col. 2, lines 65-66 & Emmelmann, [0022]) the updating/modifying of dynamic aspects of components used in HTML.

Per claim 10:

A computer program storage medium readable by a computer system having memory and encoding a computer program for executing a computer process creating ...in memory, wherein ...used by the server computer system to create server-side objects for dynamically rendering web page content, the web page content delivered to a client-side computer system and displayed as a web page on the client computer system, said computer process comprising:

(DuFresne: Col. 2, lines 58-59, "a design, development, maintenance, and deployment systems and methods...running in an HTTP network environment ...", col. 6, lines 22-23, "...server manages information stored under its control (computer program storage medium readable by a computer system)...", col. 3, lines 4-6, "deploying client-server applications involves inserting executable tags in an HTML source (dynamically rendering web page content) to a displayable page (delivered & displayed on client)", col. 6, lines 22-26, "...server manages information stored (in memory) ...runs a program to manage HTML files (in memory)...")

Art Unit: 2191

- receiving a request from the client specifying a dynamic web page content file;
- processing the dynamic web page content file to produce a source code file containing source code that represent control objects declared in the web page content file;
- compiling the source code file...from which a set of hierarchical objects can be instantiated to produce web page authoring language that produces a web page for display;
- wherein dynamic web page content file declaratorily refers to one or more additional dynamic web page content files, each reference to the one or more additional web page content files correspond to a single hierarchical object within the set of hierarchical objects.

See rejection of limitations in claim 1 above.

DuFresne failed to disclose specific details regarding 'creating a class'. However Emmelmann suggested component classes [0142] & [0189]. Therefore, it would have been obvious, to one of ordinary skill in the art to consider the components (col. 8, line 60) of the forms as disclosed by DuFresne to be of a class format, as classes are well known and commonly used in HTML programming. Both inventions provide dynamic client/server environments (DuFresne, col. 2, line 63 & Emmelmann, [0001]) and attempt to simplify (DuFresne, col. 2, lines 65-66 & Emmelmann, [0022]) the updating/modifying of dynamic aspects of components used in HTML.

Per claim 11:

In a server computer system having memory, a method of creating a plurality of web page responses having dynamically rendered web page content, the web page responses delivered to

Art Unit: 2191

one or more client-side computer systems and displayed as a web pages on the client computer systems, said method comprising:

(DuFresne: Col. 2, lines 58-59, "a design, development, maintenance, and deployment systems and methods...running in an HTTP network environment ...", col. 6, lines 22-23, "...server manages information stored under its control (computer system having memory)...", col. 3, lines 4-6, "deploying client-server applications involves inserting executable tags in an HTML source (dynamically rendering web page content) to a displayable page (delivered & displayed on client)", col. 6, lines 22-26, "...server manages information stored (in memory) ...runs a program to manage HTML files (in memory)...")

-receiving a request from the client computer system for the web page, wherein the request identifies a dynamic web page content file;

See rejection of limitations in claim 1 above.

-creating a hierarchical data model containing one or more control objects to store elements of the dynamic web page content file;

(DuFresne: Col. 3, lines 41-52, "...instructions can be nested and grouped (hierarchical objects) to form a script which performs predefined tasks...a script is embedded in a tag extension such that executing (instantiate to produce web page authoring language) the extension causes the script to expand and further execute other tag extensions or instructions...in a hypertext source (hierarchical data model), a value associated with a tag extension can be a script such that processing the source executes the tag extension and further causes the script to expand and

execute a string of tags and instructions...”, col. 4, lines 37-39, “The resulting source to the displayable page is...a pure HTML document which can be interpreted by the client browser (produces a web page for display).”)

-generating a source code file related to the dynamic web page content file based on the evaluation of the data model;

(DuFresne: col. 4, lines 37-39, “The resulting source (generated source file) to the displayable page is...a pure HTML document which can be interpreted by the client browser (produces a web page for display).”)

-compiling the source code file to create a compiled...memory;

(DuFresne: Col. 3, lines 41-52, “...instructions can be nested and grouped (hierarchical objects) to form a script which performs predefined tasks...a script is embedded in a tag extension such that executing (instantiate to produce web page authoring language) the extension causes the script to expand (compile / transform) and further execute other tag extensions or instructions...in a hypertext source, a value associated with a tag extension can be a script such that processing the source executes the tag extension and further causes the script to expand and execute a string of tags and instructions...”, col. 4, lines 37-39, “The resulting source to the displayable page is...a pure HTML document which can be interpreted by the client browser (produces a web page for display).”)

-returning a...reference to the server computer system enabling the server computer system to instantiate server-side processing objects...to dynamically generate web page content;

(DuFresne: Col. 3, lines 19-27, "a tag extension is often directed to (a class reference to) a field name corresponding to a static value...A tag extension can...be directed to a field name which corresponds to a value representing another tag extension or a URL...A value, therefore, has a dynamic property. A full execution of a tag extension requires exhaustively expanding all related values or commands until a value is static...", col. 9, lines 17-18, "processor constructs a Web page by incorporating database records on the fly (dynamically generate web page content).", col. 9, lines 25-27, "each service request by the client invokes (client returns a class reference to the server computer system) a template containing both static HTML and executable codes in an input field.", col. 9, lines 30-35, "The template is processed by the processor (instantiate server-side processing objects) and each code is replaced...or executed...This process produces an output HTML page which is displayable (dynamically generate web page content) on the client browser.")

-rendering the dynamic web page content into a web page response for delivery to the client computer system;

(DuFresne: Col. 9, lines 33-34, "This process produces an output HTML page which is displayable (rendering dynamic web page content for delivery to client) on the client browser.")

-conducting the web page response to the requesting client computer system;

(DuFresne: Col. 6, lines 18-21, "The HTML provides rules for displaying...and means for specifying connectors to allow users to traverse through the Internet...", col. 6, lines 32-33, "...and returns the page to the client (conduct response to requesting client) for display.")

-receiving a second request for the web page for the web page, wherein the request identifies a dynamic web page content file;

(DuFresne: Col. 6, line 29- 31, "When a client makes a request for a particular Web page controlled by the Web server, the server processes the request...", col. 9, lines 17-18, "the processor constructs a Web page by incorporating database records on the fly." The client requests content via a template HTML form. The server manages Web applications for databases. (Col. 8, lines 62-63, "A template is an HTML form to define contents of a display Web page requested by a client (request identifies a dynamic web page content file).", (Col. 9, lines 45-47), "Data in each element (of database) is accessed through a template and processed by the processor to construct an (dynamic) output Web page requested by the client." DuFresne allows for a repeat, a second response, as shown in FIG. 6.)

-determining that a compiled...[code] for that dynamic web page content file resides in memory;
(See rejection of limitations addressed in claim 8 above.)

-returning a...reference to the server computer system enabling the server computer system to instantiate server-side processing objects...to dynamically generate web page content;

(DuFresne: Col. 4, lines 4-8, Templates and content databases (classes) are controlled by the server. Col. 4, lines 19-29, "The template comprises a text area for entering a source code which includes hypertext codes embedded with tag extensions (returning a class reference to the server) executable by the processor. Each tag specifies, in a data record, a field name having a value such that executing the tag extension (instantiate server-side objects) replaces each tag with the corresponding value (dynamically generate content)...")

-rendering the dynamic web page content into a second web page response;

(DuFresne: Col. 9, lines 33-34, "This process produces an output HTML page which is displayable (rendering dynamic web page content for delivery to client) on the client browser." DuFresne allows for a repeat, a second response, as shown in FIG. 6.)

-conducting the second web page response to the requesting client computer system; wherein the dynamic web page content file declaratorily refers to one or more additional dynamic web page content files, each reference to the one or more additional dynamic web page content files correspond to a single hierarchical object within the set of hierarchical objects.

(DuFresne: See limitations addressed above in instant claim. DuFresne allows for a repeat, a second response, as shown in FIG. 6.)

DuFresne failed to explicitly disclose "class" formats / stored in memory. However he did suggest, col. 9, lines 30-32, "The template is processed by the processor and each code is

Art Unit: 2191

replaced by a stored data value (related class is compiled and stored in memory) or executed to perform certain task (if not stored continue with processing to compile / expand code)..."

Therefore, it would have been obvious, to one of ordinary skill in the art, at the time of the invention, to modify DuFresne's invention, to include explicit details regarding 'determining that a compiled class...resides in memory', because DuFresne did disclose retrieving code from memory else compile / expand code, and it only is logical to first determine whether stored code exists before continuing to process. It would have been obvious, to one of ordinary skill in the art, at the time of the invention to consider the hypertext environment (col. 2, line 60) as described by DuFresne to include class formats stored and compiled because (col. 4, lines 5-8) "filed names and corresponding contents of a template are stored in a content database. Both the template and content databases are controlled by the server of the present invention." The use of class format is well known in the client server applications.

Per claim 12:

A computer program storage medium readable by a computer system having memory and encoding a computer program for executing a computer process creating a plurality of web page responses having dynamically rendered web page content, the web page responses delivered to one or more client-side computer systems and displayed as a web pages on the client computer systems, said computer process comprising:

(DuFresne: Col. 2, lines 58-59, "a design, development, maintenance, and deployment systems and methods...running in an HTTP) network environment (data signal embodied in a carrier

Art Unit: 2191

wave by a system)...” (emphasis added), col. 6, lines 22-26, “...server manages information stored (in memory) ... runs a program to manage HTML files (class in memory)...”, col. 6, lines 22-23, “...server manages information stored under its control (computer program storage medium / encoding a computer program).)

- receiving a request from the client computer system for the web page, wherein the request identifies a dynamic web page content file;
- creating a hierarchical data model containing one or more control objects to store elements of the dynamic web page content file;
- generating a source code file related to the dynamic web page content file based on the evaluation of the data model;
- compiling the source code file to create a compiled...in memory;
- returning a...reference to the server computer system enabling the server computer system to instantiate server-side processing objects from that class to dynamically generate web page content;
- rendering the dynamic web page content into a web page response for delivery to the client computer system; conducting the web page response to the requesting client computer system;
- receiving a second request for the web page for the web page, wherein the request identifies a dynamic web page content file;
- determining that a compiled...[code] for that dynamic web page content file resides in memory;

- returning a...reference to the server computer system enabling the server computer system to instantiate server-side processing objects from that class to dynamically generate web page content;
- rendering the dynamic web page content into a second web page response; and conducting the second web page response to the requesting client computer system;
- wherein the dynamic web page content file declaratorily refers to one or more additional dynamic web page content files, each reference to the one or more additional dynamic web page content files correspond to a single hierarchical object within the set of hierarchical objects.

This is a "computer program storage medium readable by a computer system having memory and encoding a computer program" version of claim 11 above. See rejection of limitations in claim 11.

DuFresne failed to disclose specific details regarding 'creating a class'. However Emmelmann suggested component classes [0142] & [0189]. Therefore, it would have been obvious, to one of ordinary skill in the art to consider the components (col. 8, line 60) of the forms as disclosed by DuFresne to be of a class format, as classes are well known and commonly used in HTML programming. Both inventions provide dynamic client/server environments (DuFresne, col. 2, line 63 & Emmelmann, [0001]) and attempt to simplify (DuFresne, col. 2, lines 65-66 & Emmelmann, [0022]) the updating/modifying of dynamic aspects of components used in HTML.

Per claim 13:

Art Unit: 2191

A computer data signal embodied in a carrier wave by a computing system having memory and encoding a computer program for executing a computer process creating a plurality of web page responses having dynamically rendered web page content, the web page responses delivered to one or more client-side computer systems and displayed as a web pages on the client computer systems, said computer process comprising:

(DuFresne: Col. 2, lines 58-59, "a design, development, maintenance, and deployment systems and methods...running in an HTTP) network environment (data signal embodied in a carrier wave by a system)...” (emphasis added), col. 6, lines 22-26, "...server manages information stored (in memory) ...runs a program to manage HTML files (class in memory)...”, col. 6, lines 22-23, "...server manages information stored under its control.)

- receiving a request from the client computer system for the web page, wherein the request identifies a dynamic web page content file;
- creating a hierarchical data model containing one or more control objects to store elements of the dynamic web page content file;
- generating a source code file related to the dynamic web page content file based on the evaluation of the data model;
- compiling the source code file to create a compiled class in memory;
- returning a...reference to the server computer system enabling the server computer system to instantiate server-side processing objects from that class to dynamically generate web page content;

- rendering the dynamic web page content into a web page response for delivery to the client computer system;
- conducting the web page response to the requesting client computer system;
- receiving a second request for the web page for the web page, wherein the request identifies a dynamic web page content file;
- determining that a compiled...for that dynamic web page content file resides in memory; returning a...reference to the server computer system enabling the server computer system to instantiate server-side processing objects from that class to dynamically generate web page content;
- rendering the dynamic web page content into a second web page response;
- conducting the second web page response to the requesting client computer system;
- wherein the dynamic web page content file declaratorily refers to one or more additional dynamic web page content files, each reference to the one or more additional dynamic web page content files correspond to a single hierarchical object within the set of hierarchical objects.

This is a "computer data signal embodied in a carrier wave by a computing system computer program storage medium readable by a computer system having memory and encoding a computer program" version of claim 11 above. See rejection of limitations in claim 11.

DuFresne failed to disclose specific details regarding 'creating a class'. However Emmelmann suggested component classes [0142] & [0189]. Therefore, it would have been obvious, to one of ordinary skill in the art to consider the components (col. 8, line 60) of the forms as disclosed by

Art Unit: 2191

DuFresne to be of a class format, as classes are well known and commonly used in HTML programming. Both inventions provide dynamic client/server environments (DuFresne, col. 2, line 63 & Emmelmann, [0001]) and attempt to simplify (DuFresne, col. 2, lines 65-66 & Emmelmann, [0022]) the updating/modifying of dynamic aspects of components used in HTML.

Per claim 14:

A computer program product encoding a computer program for executing in a computer system a computer process for creating...in memory, wherein...is used by a server computer system to create server-side objects for dynamically rendering authoring language elements, the elements are delivered to a client-side computer system and processed on the client computer system, said process comprising:

- receiving a request from the client computer system for the resource, wherein the request identifies a dynamic web page resource;
- processing the resource to generate a source code file related to the resource;
- compiling the source code file to create a compiled...in memory to enable the instantiation of hierarchical objects of the compiled...;
- wherein the source code file declaratorily refers to one or more dynamic web page content files, each reference to the one or more dynamic web page content files correspond to a single hierarchical object within the hierarchical objects.

See rejection of limitations in claim 1 above.

DuFresne failed to disclose specific details regarding 'creating a class'. However Emmelmann suggested component classes [0142] & [0189]. Therefore, it would have been obvious, to one of ordinary skill in the art to consider the components (col. 8, line 60) of the forms as disclosed by DuFresne to be of a class format, as classes are well known and commonly used in HTML programming. Both inventions provide dynamic client/server environments (DuFresne, col. 2, line 63 & Emmelmann, [0001]) and attempt to simplify (DuFresne, col. 2, lines 65-66 & Emmelmann, [0022]) the updating/modifying of dynamic aspects of components used in HTML.

Per claim 15:

A computer program product encoding a computer program for executing in a computer system a computer process for creating a...memory as defined in claim 15, wherein the processing step of creating a data model comprises:

-parsing the resource to separate the resource into logical elements and identify relationships between the logical elements;

(DuFresne: Col. 4, lines 32-38, "The source code defined within the template is then processed (processing the dynamic web page content file) by the processor. Such process includes executing the tag extensions in the source to replace each extension (parse content file, find tag extensions) with the corresponding value so as to configure the page with the remaining hypertext codes (store into HTML data model). The resulting source to the displayable page is, therefore, a pure HTML document...")

Art Unit: 2191

-creating a plurality of hierarchically related data structures forming a hierarchical data model;
(DuFresne: Col. 3, lines 36-40, "...a tag extension in a source to a Web page is associated with a value in a database. Such a value can be static data or a variable, such as another tag extension ", col. 3, lines 50-52, "processing (creating a plurality of hierarchically related data structures) the source executes the tag extension and further causes the script to expand and execute a string of tags and instructions...(forming HTML code which is a hierarchical data model)")

-storing portions of the resource in the data structures.

See rejection of limitations addressed in claim 8 above.

DuFresne failed to disclose specific details regarding 'creating a class'. However Emmelmann suggested component classes [0142] & [0189]. Therefore, it would have been obvious, to one of ordinary skill in the art to consider the components (col. 8, line 60) of the forms as disclosed by DuFresne to be of a class format, as classes are well known and commonly used in HTML programming. Both inventions provide dynamic client/server environments (DuFresne, col. 2, line 63 & Emmelmann, [0001]) and attempt to simplify (DuFresne, col. 2, lines 65-66 & Emmelmann, [0022]) the updating/modifying of dynamic aspects of components used in HTML.

Per claim 16:

A computer program product encoding a computer program for executing in a computer system a computer process for creating...in memory as defined in claim 15, wherein the processing step comprises the following steps:

Art Unit: 2191

-performing a first analysis of the resource to generate source code related to variable declaration information;

-performing a second analysis of the resource to generate source code related to control object information;

-performing a third analysis of the resource to generate source code related to rendering information;

-storing the source code in the source code file.

See rejection of limitations as addressed in claims 6-8 above.

DuFresne failed to disclose specific details regarding 'creating a class'. However Emmelmann suggested component classes [0142] & [0189]. Therefore, it would have been obvious, to one of ordinary skill in the art to consider the components (col. 8, line 60) of the forms as disclosed by DuFresne to be of a class format, as classes are well known and commonly used in HTML programming. Both inventions provide dynamic client/server environments (DuFresne, col. 2, line 63 & Emmelmann, [0001]) and attempt to simplify (DuFresne, col. 2, lines 65-66 & Emmelmann, [0022]) the updating/modifying of dynamic aspects of components used in HTML.

Per claim 17:

A computer program product encoding a computer program for executing in a computer system a computer process for creating...in memory as defined in claim 16, wherein the processing step of generating source code comprises further comprises the step of generating an intermediate data structure, wherein the source code is generated from the intermediate data structure.

DuFresne failed to disclose specific details regarding 'creating a class'. DuFresne failed to disclose "intermediate data structures".

However, Emmelmann suggested component classes [0142] & [0189]. Emmelmann disclosed an intermediate data structure: [0122], "The ISSC (interactive server side component) processor then reads the component page...The result is an abstract syntax tree (AST)" (intermediate data structure). Also, [0123], "Each node representing a component with content in turn contains a cb-list that represents the content of the component. Also, [0129], "Fig. 9 is a flow chart of the page generation (source code generated) algorithm. It is a recursive algorithm taking a cb-list as parameter. The cb-list is processed node by node..."

Therefore, it would have been obvious, to one of ordinary skill in the art to consider the components (col. 8, line 60) of the forms as disclosed by DuFresne to be of a class format, as classes are well known and commonly used in HTML programming. Both inventions 'compile' from a first source to a second source. Emmelmann shows an intermediate compilation step, the intermediate data structure, well known in the art as language neutral. Both inventions provide dynamic client/server environments (DuFresne, col. 2, line 63 & Emmelmann, [0001]) and attempt to simplify (DuFresne, col. 2, lines 65-66 & Emmelmann, [0022]) the updating/modifying of dynamic aspects of components used in HTML.

Per claim 18:

A computer program product encoding a computer program for executing in a computer system a computer process for creating...in memory as defined in claim 16, wherein the processing step of generating an intermediate data structure further comprises:

- performing a first analysis of the resource to generate intermediate data structure elements related to variable declaration information;
- performing a second analysis of the resource to generate intermediate data structure elements related to control object information;
- performing a third analysis of the resource to generate intermediate data structure elements related to rendering information;
- generating source code from the intermediate data structure.

DuFresne disclosed a ‘plurality’ of analysis. See FIG. 6, where drawing loops back to accept more client input. DuFresne disclosed variable declaration information, control object information, and rendering information as noted in rejection of claims 6 and 7 above.

DuFresne failed to disclose a ‘class’ format. DuFresne failed to disclose “intermediate data structures”.

However, Emmelmann suggested component classes [0142] & [0189]. Emmelmann disclosed an intermediate data structure: [0122], “The ISSC (interactive server side component) processor then reads the component page...The result is an abstract syntax tree (AST)” (intermediate data structure). Also, [0123], “Each node representing a component with content in turn contains a cb-list that represents the content of the component. Also, [0129], “Fig. 9 is a flow chart of the

page generation (source code generated) algorithm. It is a recursive algorithm taking a cb-list as parameter. The cb-list is processed node by node..."

Therefore, it would have been obvious, to one of ordinary skill in the art to consider the components (col. 8, line 60) of the forms as disclosed by DuFresne to be of a class format, as classes are well known and commonly used in HTML programming. Both inventions 'compile' from a first source to a second source. Emmelmann shows an intermediate compilation step, the intermediate data structure, well known in the art as language neutral. Both inventions provide dynamic client/server environments (DuFresne, col. 2, line 63 & Emmelmann, [0001]) and attempt to simplify (DuFresne, col. 2, lines 65-66 & Emmelmann, [0022]) the updating/modifying of dynamic aspects of components used in HTML.

Per claim 19:

A computer program product encoding a computer program for executing in a computer system a computer process for creating...in memory as defined in claim 18, wherein the intermediate data structure is a generic description that may be translated into a plurality of source code language files, wherein at least one source code file is different from another source code language file.

DuFresne disclosed "a computer program product encoding a computer program for executing in a computer system a computer process for creating [code]...in memory, specifically HTML code suitable for rendering dynamic content on a client browser. DuFresne failed to disclose a 'class' format or and 'intermediate data structure'.

Art Unit: 2191

However Emmelmann suggested component classes [0142] & [0189]. Emmelmann disclosed an intermediate data structure from which different source code files may be derived: [0075], "A component can be programmed in any programming language..."

Therefore, it would have been obvious, to one of ordinary skill in the art to consider the components (col. 8, line 60) of the forms as disclosed by DuFresne to be of a class format, as classes are well known and commonly used in HTML programming. Both inventions 'compile from a first source to a second source. Emmelmann shows an intermediate compile step, the intermediate data structure, well known in the art as language neutral. Both inventions provide dynamic client/server environments (DuFresne, col. 2, line 63 & Emmelmann, [0001]) and attempt to simplify (DuFresne, col. 2, lines 65-66 & Emmelmann, [0022]) the updating/modifying of dynamic aspects of components used in HTML.

Conclusion

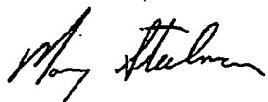
5. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mary Steelman, whose telephone number is (571) 272-3704. The examiner can normally be reached Monday through Thursday, from 7:00 AM to 5:30 PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached at (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

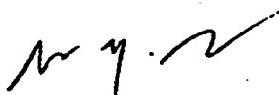
Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mary Steelman



03/24/2005



WEI Y. ZHEN
PRIMARY EXAMINER

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.